UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

In re EXPRESS MOBILE CASES	Case Nos. 3:19-cv-06559-RS 3:20-cv-06152-RS 3:20-cv-08297-RS 3:20-cv-08321-RS 3:20-cv-08335-RS 3:20-cv-08339-RS 3:20-cv-08461-RS 3:20-cv-08491-RS 3:20-cv-08492-RS 3:21-cv-01145-RS 3:21-cv-02001-RS **DECLARATION OF CHRISTOPHER SCHMANDT IN SUPPORT OF DEFENDANTS' CLAIM CONSTRUCTION BRIEF**

I, Christopher Schmandt, declare as follows:

1. I submit this declaration in support of the responsive claim construction brief submitted by X.Commerce, Inc., Adobe Inc., Wix.com, Ltd and Wix.com, Inc. (collectively "Wix"), Microsoft Corporation, LinkedIn Corporation, Adobe Inc., Amazon.com, Inc., Pinterest, Inc., Oath Holdings, Inc., SAP America, Inc., SAP Labs, LLC, SAP S.E. (collectively "SAP"), Booking.com B.V., Salesforce.com, Inc., Dropbox, Inc., and Slack Technologies, Inc. (collectively "Defendants").

2. I make this declaration based on my own personal knowledge, information and belief, and the materials set forth below. I would and could competently testify to the matters set forth herein if called upon to do so.

3. I reserve the right to supplement or amend this declaration based on any new information that is received and is relevant to my opinions, included any additional claim construction briefing from Express Mobile, Inc. ("Express Mobile"), and any declarations or other testimony from any witness testifying on Express Mobile's behalf.

4. I further reserve the right to supplement or amend this declaration in view of any relevant order of the Court or any papers served or filed by Express Mobile in this case, or in any other case brought by Express Mobile relating to the three asserted patents discussed in this declaration.

I. QUALIFICATIONS

5. A copy of my curriculum vitae is attached as Appendix A. Below, I provide a brief summary of qualifications relevant to my opinions set forth in this declaration.

6. I received my B.S. degree in Electrical Engineering and Computer Science from M.I.T. in 1978, and my M.S. also from M.I.T., in Visual Studies (Computer Graphics). I had been employed at M.I.T. since 1980 until my retirement in 2018, initially at the Architecture Machine Group, which was an early computer graphics research laboratory. In 1985, I helped found the Media Laboratory and continue to work there to date. My research spans distributed communication and collaborative systems, with an emphasis on multi-media and user interfaces. I have over 70 published conference and journal papers and one book in these fields.

7. Until my recent retirement, I was employed as a Principal Research Scientist at the Media Laboratory at M.I.T. In that role, I also served as faculty for the M.I.T. Media Arts and Sciences

academic program. I have 40 years of experience in the field of Media Technology, and was a founder of the M.I.T. Media Laboratory.

8. Most recently in my previous role, I ran a research group titled "Living Mobile." Students under my direct supervision and I have built client software on desktop computers, wearable computers, PDAs, mobile phones, pagers, and other ad hoc devices of our own design. In my faculty position, I taught courses and directly supervised student research and theses at the Bachelors, Masters, and PhD level. The research that I supervised included the use of Java to create applications and applets. I oversaw the Masters and PhD thesis programs for the entire Media Arts and Sciences academic program. I also served on the Media Laboratory intellectual property committee for many years.

9. I have personally used many of the technologies at issue in this case and feel comfortable with rendering technical opinions relating to the operation of such technologies. For example, I have written HTML code to directly generate web pages and I have also written code generators to programmatically generate HTML documents. I have also written parsers to interpret HTML code. I have set up and used various databases on a variety of occasions to store data, and I have also designed and coded a number of databases of my own.

10. I have been familiar with the World Wide Web since shortly after its creation in the early 1990s. I am familiar with the creation of web sites and web pages, including those that are dynamically created using a database, by personally engaging in website development and in the context of my experience as a businessman and also as an academic researcher. I have founded two companies and have advised another—all of these companies have engaged in web site development in one form or another. I have also directly taught and worked with numerous individuals who have engaged in web site development and the use of What You See Is What You Get ("WYSIWYG") developer tools, database technology, and virtual machine programming languages.

11. Since the mid-1980s, my work was heavily oriented around client-server software architectures for distributed computing, both to handle low-level resource management such as files and media, and higher-level aspects such as providing applications and user interfaces to thin clients such as mobile phones.

12. In the late 1980s and early 1990s, I served on a Working Group reporting to the Internet Activities Board (IAB) which later became the Internet Engineering Task Force (IETF) on network requirements for multimedia computing, which required extensive knowledge of various Internet protocols.

II. COMPENSATION AND PRIOR TESTIMONY

13. I am being paid my standard consulting rate of $500 per hour for my time spent working on this case. I am also being reimbursed for any expenses I incur (e.g., travel expenses). My compensation is not contingent on any findings or opinions herein, and I will be paid for my services regardless of the outcome. Thus, no part of my compensation is dependent upon the outcome of this case.

14. In the past four years, I have been deposed or testified at trial and IPR proceedings in a number of cases representing both plaintiffs and defendants. These prior proceedings are detailed in Appendix A attached to this declaration.

III. MATERIALS CONSIDERED

15. In forming the opinions I express in this declaration, I have reviewed U.S. Patent No. 9,063,755 ("the '755 patent"), U.S. Patent No. 9,471,287 ("the '287 patent"), and U.S. Patent No. 9,928,044 ("the '044 patent"), the file histories for the '755, '287, and '044 patents, the documents to which these patents claim priority, and all the documents I cite to or otherwise reference in this declaration, including Express Mobile's Brief ("Opening Brief")[1] and the Declaration of Glenn Weadock in support of Express Mobile's Opening Brief ("Weadock Decl.").[2]

IV. LEGAL UNDERSTANDING

16. I am not an attorney, but counsel has explained to me the following legal principles, which I applied in conducting the analyses expressed in this declaration.

A. General Principles of Claim Construction

17. I understand that the purpose of claim construction is to give claim terms the meaning understood by a person having ordinary skill in the art ("POSA" or "ordinary artisan") at the time of the

[1] *Express Mobile, Inc. v. Wix.com*, 3:19-cv-06559-RS, Dkt. 142.

[2] *Express Mobile, Inc. v. Wix.com*, 3:19-cv-06559-RS, Dkt. 142-1.

claimed invention, when considered in the context of the patent. For the purposes of this declaration, I have been asked to assume that the time of the purported inventions claimed in the ’755 patent, the ’287 patent, and the ’044 patent is November 11, 2008. I understand that in other litigations involving the ’755, ’287, and ’044 patents, Express Mobile has contended that the asserted claims are entitled to an earlier date of invention in January 2008. My opinions set forth in this declaration would not change if the time of the claimed invention was determined to be January 2008 instead of November 11, 2008.

18. I understand that claim terms are interpreted in the context of not only the claim in which the disputed term appears, but in the context of the entire patent, including the specification. I also understand that the history of that patent’s prosecution before the U.S. Patent & Trademark Office (“PTO”) is relevant to the meaning of the claims.

19. To that end, I understand that that claims are not to be interpreted in a vacuum and must be considered within the context of this intrinsic evidence. I understand that this “intrinsic evidence” includes the subject patent itself, the history of that patent’s prosecution before the PTO, patents (or other materials) incorporated by reference into the subject patent, and patents (or other materials) cited to or by the PTO during the prosecution of the subject patent. Additionally, I understand that the patentee can act as his or her own “lexicographer” and provide his or her own special definition of a claim term, even if the patentee’s definition is different than the ordinary usage of that term. When a patentee clearly acts as his or her own lexicographer, I understand that the definition provided by the patentee should be applied.

20. I further understand that dictionary definitions and other extrinsic evidence, such as expert testimony, may be considered to the extent it aids in the understanding claim terms, but extrinsic evidence may not be used to vary the meaning given to a claim term based on the intrinsic evidence.

B. Indefiniteness

21. I understand that a claim term may render the claim in which it appears indefinite and therefore invalid, if a POSA cannot determine the scope of the claim with reasonably certainty, claim based on the claim language, the specification, the prosecution history, and the knowledge in the relevant art.

V. OVERVIEW OF THE ’755, ’287, AND ’044 PATENTS

22. The ’755 patent is entitled “Systems and Methods for Presenting Information on Mobile Devices.” The titles for the ’287 and ’044 patents are similar. The ’287 patent is entitled “Systems and Methods for Integrating Widgets on Mobile Devices.” The ’044 patent is entitled “Systems and Methods for Programming Mobile Devices.” The ’755 patent was filed on April 6, 2009. The ’287 and ’044 patents are both continuations of the ’755 patent and share a substantively identical specification. All three are directed to “generating and distributing programming to mobile devices over a network.”[3] Each patent’s Abstract states that “Devices are provided with Players specific to each device and Applications that are device independent. Embodiments include a full-featured WYSIWYG authoring environment, including the ability to bind web components to objects.”[4] Because the specifications are substantively identical, unless material differences exist, citations below when referring to the ’755, ’287, and ’044 patents are only to the ’755 patent. Similar disclosures can be found in the ’287 and ’044 patents.

23. The purported invention disclosed is an authoring tool that allows a designer to create Applications and Players that integrate with third-party web services. The invention is generally directed to two concepts: (1) an authoring tool that allows a user to customize the display for receiving inputs and displaying outputs of web services; and (2) an authoring tool that generates a device-independent Application and a device-specific Player that embody this user-customized display for use by another on a client device. Because the user of the authoring tool may be different than the ultimate end-user of the Application and Player, I will refer to the former as the “creator” and the latter as the “end-user” or “client.” Each of these concepts is discussed below.

24. The first general concept of the ’755, ’287, and ’044 patents is the idea of allowing a creator to customize the display by matching UI objects to inputs and outputs of web services. As described in the specification:

> Web service 230 is a plurality of services obtainable over the Internet. Each web service is identified and/or defined as an entry in web component registry 230, which is a database, XML file, or PDL that exists on a computer that may be a server previously described or another server 120. Web component registry 230 is provided through server 120 to authoring platform 110 so that a user of the authoring platform may bind web

[3] ’755 patent at Abstract; ’287 patent at Abstract (same); ’044 patent at Abstract (same).

[4] ’755 patent at Abstract; ’287 patent at Abstract (same); ’044 patent at Abstract (same).

> services 230 to elements to be displayed on device 130, as described subsequently.
>
> . . . .
>
> Authoring platform 110 further permits a user of the authoring platform to associate objects, such as objects for presenting on screen 137, with components of one or more web services 230 that are registered in web component registry 220. In one embodiment, information is provided in an XML file to web component registry 220 for each registered components of each web service 230. Web component registry 220 may contain consumer inputs related to each web service 230, environmental data such as PIM, time or location values, persistent variable data, outputs related to the web service, and/or optional hinting for improving the user's productivity.
>
> A user of authoring platform 110 of system 200 may define associations with web services as WebComponent Bindings. In one embodiment, authoring platform 110 allows a user to associate certain objects for display that provide input or output to components of web service 230. The associated bindings are saved as a PDL in server 120.[5]

25. Stated differently, and as described in the ’755 patent at 8:18–53, the purported invention allows the creator to choose various web services it would like to integrate into an Application, and choose the UI object that will be used to receive inputs or display outputs. For example, the creator may want to display an RSS feed.[6] The creator would select the RSS feed she wants from the registry of feeds available.[7] The creator could then bind the outputs of that web service (e.g. the “item-description” shown at XMO_00002266) to any UI object that is useful to display that datatype.[8] So for example, the “item-description” shown on XMO_00002266 is of datatype “multiLineString” as annotated below:[9]

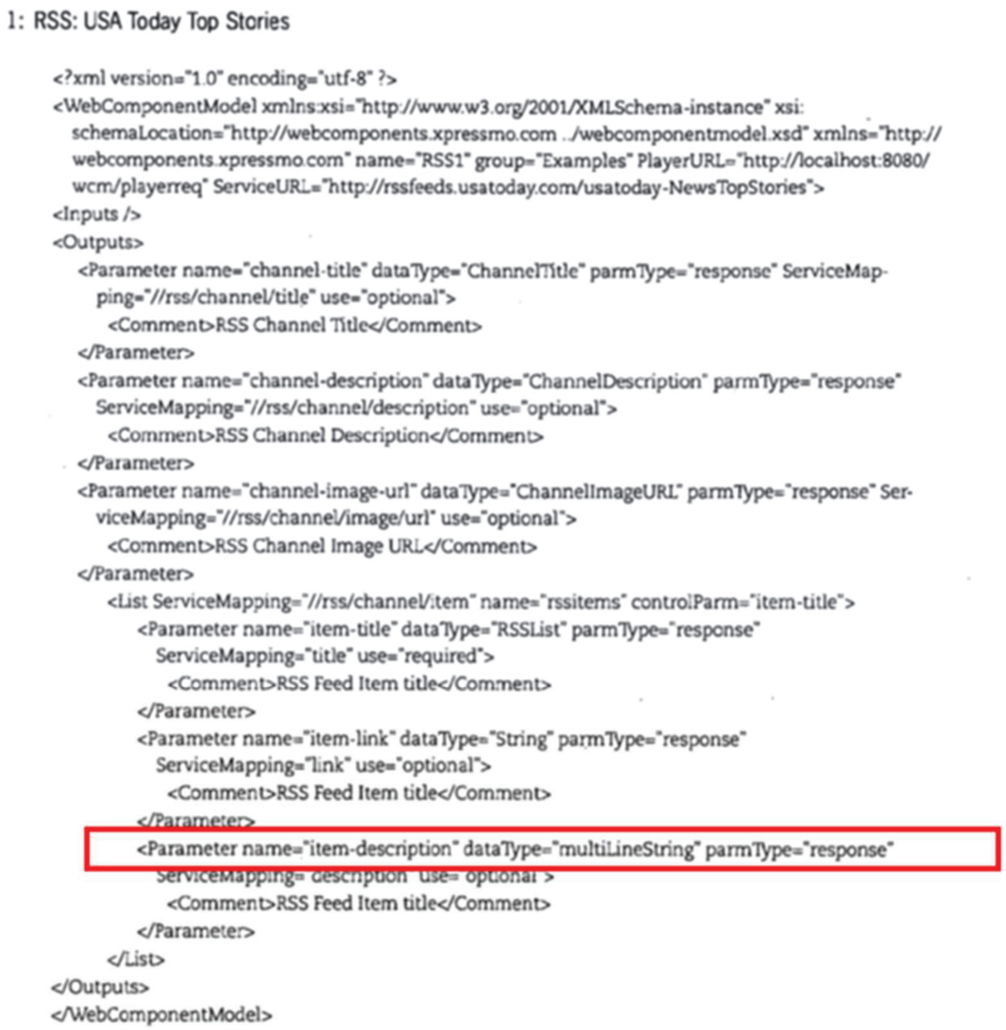
[5] ’755 patent at 8:18–53; *see also* Exh. 5, Excerpt from 9,063,755 File History, Reply “B” Under 37 C.F.R. §1.116(e), May 30, 2013 at 4 (quoting paragraphs [0066] and [0069], corresponding to the first two paragraphs above) (XMO_00002851).

[6] *See* ’755 patent at 9:17-26. “RSS” stands for Real Simple Syndication and an RSS feed provides subscribers to the feed with updates to the associated website in a standardized format.

[7] *See* Exh. 1, Excerpt from 9,063,755 File History, U.S. Prov. Pat. Appl. No. 61/113,471, Nov. 11, 2008 at XMO_00002266 (showing a “USA Today Top Stories” RSS feed in the web component registry).

[8] *See* ’755 patent at 9:17-26.

[9] *See* Exh. 1, Excerpt from 9,063,755 File History, U.S. Prov. Pat. Appl. No. 61/113,471, Nov. 11, 2008 at XMO_0002266.



1: RSS: USA Today Top Stories

```
<?xml version="1.0" encoding="utf-8" ?>
<WebComponentModel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
  schemaLocation="http://webcomponents.xpressmo.com ../webcomponentmodel.xsd" xmlns="http://
  webcomponents.xpressmo.com" name="RSS1" group="Examples" PlayerURL="http://localhost:8080/
  wcm/playerreq" ServiceURL="http://rssfeeds.usatoday.com/usatoday-NewsTopStories">
<Inputs />
<Outputs>
  <Parameter name="channel-title" dataType="ChannelTitle" parmType="response" ServiceMap-
    ping="//rss/channel/title" use="optional">
     <Comment>RSS Channel Title</Comment>
  </Parameter>
  <Parameter name="channel-description" dataType="ChannelDescription" parmType="response"
    ServiceMapping="//rss/channel/description" use="optional">
     <Comment>RSS Channel Description</Comment>
  </Parameter>
  <Parameter name="channel-image-url" dataType="ChannelImageURL" parmType="response" Ser-
    viceMapping="//rss/channel/image/url" use="optional">
     <Comment>RSS Channel Image URL</Comment>
  </Parameter>
     <List ServiceMapping="//rss/channel/item" name="rssitems" controlParm="item-title">
        <Parameter name="item-title" dataType="RSSList" parmType="response"
          ServiceMapping="title" use="required">
           <Comment>RSS Feed Item title</Comment>
        </Parameter>
        <Parameter name="item-link" dataType="String" parmType="response"
          ServiceMapping="link" use="optional">
           <Comment>RSS Feed Item title</Comment>
        </Parameter>
        <Parameter name="item-description" dataType="multiLineString" parmType="response"
          ServiceMapping="description" use="optional">
           <Comment>RSS Feed Item title</Comment>
        </Parameter>
     </List>
</Outputs>
</WebComponentModel>
```

As shown in Table 1 of the ’755 patent, “multiLineString” can be associated with a “Text Area” UI object or a “Paragraph” UI object. This allows the creator to customize how the output is displayed, whether it is as a text area UI object or as a paragraph UI object.

26. The second general concept of the ’755, ’287, and ’044 patents is the idea of separating the code generated by the authoring tool into a device-independent Application and a device-dependent Player and providing those codes to the end-user’s device. As stated by the applicant during prosecution:

> In one aspect of the present invention, the device-independent Application and device-dependent Player are both provided to a device, such as a smartphone. Both codes are executed or interpreted on the same device, which communicates with a web service. By partitioning code for accessing web services into an Application and a Player has an advantage for maintaining websites. Thus, for example, code that is device specific may be maintained separately from code that web service specific. If a new device comes on the market, or if it is found that the device-dependent code for a specific device needs an update, a new Player is developed and provided to those specific devices. Likewise, if new capabilities are

> provided for certain web services, or if a new web service is available, then a new Application is developed for those web services, and provided to devices requesting those services.
>
> As one example of the operation of the Application and Player on a device, the device receives an executable code in the form of a Player. The device also receives an Application that is a device-independent code containing web service information. The execution of the Player on the device causes the Application to be interpreted and the following actions to occur on or in communication with the device: a display requesting web service input is provided, a web service request is generated and provided to the web service, the resulting web service output is received, and a display representative of the output is generated.[10]

27. The specification claims the advantage of this system is that "all of the device-dependent programming is provided to the device only once (or possibly for some small number of upgrades), permitting a smaller Application, which is the same for each device."[11]

VI. SKILL LEVEL OF PERSON HAVING ORDINARY SKILL IN THE ART

28. I have previously offered opinions in cases involving the '755, '287, and '044 patents that a POSA in the subject matter of these patents would have at least a Bachelor of Science in Computer Science and approximately three years of experience as a web developer. Such a person of ordinary skill would be well versed in HTML, JavaScript, and CSS and would be familiar with server-side scripting and developing programs using higher level languages, such as Java, C++, or Visual Basic. In forming the opinion set forth below, I have reviewed the Declaration of Glenn Weadock ("Weadock Decl."), served on October 22, 2021 by Express Mobile in support of its opening claim construction brief. At paragraph 17 of Mr. Weadock's declaration, he provides his opinions regarding the qualification of a POSA. To the extent there are any differences between Mr. Weadock and myself with respect to the qualifications of a POSA, I do not believe they are significant and would not affect any of my opinions set forth below.

[10] Exh. 8, Excerpt from 9,063,755 File History, Appeal Brief filed under C.F.R. §1.192, Nov. 26, 2014 at 8 (XMO_00001784 at XMO_00002998).

[11] '755 patent at 5:51-55.

VII. DISPUTED TERMS OF THE '755, '287, AND '044 PATENTS

A. Application/application

Express Mobile's Proposed Construction	Defendants' Proposed Construction
device-independent code containing instructions for a device	device independent code that is separate from the Player/player and is interpreted or executed by the Player/player

29. In the context of the '755, '287, and '044 patents, it is my opinion that a POSA would recognize that the claimed "Application" or "application" (hereinafter, "Application") refers to device-independent code that is separate from the claimed "Player" or "player" (hereinafter, "Player"), and is interpreted or executed by the Player.

30. Mr. Weadock agrees that the claimed Application is device-independent code. Mr. Weadock argues, however, that the claimed Application need not be separate from the claimed Player or interpreted or executed by the Player. I disagree with Mr. Weadock because in my opinion, a POSA would understand that a given set of device-independent code must satisfy both of these requirements, consistent with Defendants' proposed construction.

31. ***First,*** after reading the claims and the prosecution history, a POSA would recognize that the Application must be separate from the Player. The asserted claims refer to the Player and Application as distinct claim elements, which suggests they are separate sets of code. Mr. Weadock opines that because the Application and Player are recited separately in the claims, no further separateness is required. Weadock Decl. ¶ 28. I disagree. A POSA would recognize that for device-independent code to be considered the claimed "Application," it must be a separate set of code from the Player, based on the prosecution history.

32. From my review of the prosecution history of the '755 patent, it is clear that the applicant specifically distinguished the claimed invention over prior art on the basis of this "separate" code requirement. For example, the applicant argued that there "is no teaching or suggestion in McCain [the prior art] of an authoring tool that provides two separate codes: a device-dependent code (such as the claimed Player) and device-independent code (such as the claimed Application)."[12] As the applicant

[12] Exh. 4, Excerpt from 9,063,755 File History, Amendment "A" Under 37 C.F.R. §1.111, Mar. 6, 2013 at 11-13 (XMO_00001784 at XMO_00002804-2806).

explained, "the claimed invention, in contrast, operates by partitioning the code required for functionality into device-independent code and device-dependent code."[13] According the applicant, "[b]y partitioning code for accessing web services into an Application and a Player has an advantage for maintaining websites." [sic][14]

33. For the Application to be separate and partitioned from the Player as applicants were using the terms during prosecution, a POSA would recognize that the Application and Player must be entirely different sets of code. Again, this is reflected in the applicants' characterization of McCain, which applicants argued "clearly teaches combining all code (executable, device dependent code[)] as well as parameters (device-independent code) into C2 components, which are then provided to the browser to generate a display."[15] Or, as applicants succinctly told the Examiner, "[t]he solution of ***McCain*** is one code that is delivered to the device, where it is executed by a browser on the device."[16] Applicants further emphasized that: "<u>There is no teaching or suggestion in ***McCain*** of separating binary, non-binary, or executable components into different codes.</u>"[17]

34. The applicants further explained the alleged advantages provided by the separate Application and Player in their appeal brief submitted during prosecution:

> [P]artitioning code for accessing web services into an Application and a Player has an advantage for maintaining websites. Thus, for example, code that is device specific may be maintained separately from code that [is] web service specific. If a new device comes on the market, or if it is found that the device-dependent code for a specific device needs an update, a new Player is developed and provided to those specific devices. Likewise, if new capabilities are provided for certain web services, or if a new web service is available, then a new Application is developed for those web services, and provided to devices requesting those services.[18]

[13] Exh. 7, Excerpt from 9,063,755 File History, Reconsideration after Non-Final Rejection, Jan. 16, 2014 at 13 (XMO_00001784 at XMO_00002938).

[14] Exh. 8, Excerpt from 9,063,755 File History, Appeal Brief filed under C.F.R. §1.192, Nov. 26, 2014 at 8 (XMO_00001784 at XMO_00002998).

[15] Exh. 4, Excerpt from 9,063,755 File History, Amendment "A" Under 37 C.F.R. §1.111, Mar. 6, 2013 at 10 (XMO_00001784 at XMO_00002803).

[16] Exh. 4, Excerpt from 9,063,755 File History, Amendment "A" Under 37 C.F.R. §1.111, Mar. 6, 2013 at 9 (XMO_00001784 at XMO_00002802).

[17] *Id.* at 10 (emphasis in original).

[18] Exh. 8, Excerpt from 9,063,755 File History, Appeal Brief filed under C.F.R. §1.192, Nov. 26, 2014 at 8 (XMO_00001784 at XMO_00002998).

35. A POSA would recognize that applicants were describing the claimed Application and Player as entirely separate sets of code, so that one or the other could be changed, debugged, updated, and sent independently of one another. For example, once an Application is developed, it need not be changed or modified for different device architectures. Similarly, once a Player is developed for a particular device architecture, it need not be updated each time an Application is changed to incorporate new features, or each time a new Application becomes available.

36. Mr. Weadock opines that the claimed Application is "not necessarily separate" from the Player, because the Application can allegedly "be an integrated whole with the 'Player.'"[19] But Mr. Weadock's opinion in this regard is contradicted by the portions of the file history discussed above, in which the applicants repeatedly argued that the claimed Application and Player are "separate" from one another and not packaged together in a single set of code as was done in the prior art to McCain.

37. In support of his opinion that the Application need not be separate from the Player, Mr. Weadock cites to two passages of the specification that discuss how certain aspects of the Application logic can be found on the server.[20] These excerpts, however, actually support Defendants' proposed construction and demonstrate that the Application and Player are indeed separate sets of code. The first excerpt states:

> In one embodiment, the architecture of Player P includes an abstraction interface that separates all device, operating systems and virtual machine dependencies from the Player's Application model business logic (that is, the logic of the server-side facilities) that extend the Application on the Player so that it is efficiently integrated into a comprehensive client/server Application.[21]

38. As an initial matter, I note that nothing in this excerpt refers to an Application "integrated whole with the Player" as Mr. Weadock suggests. While this passage is poorly written, a POSA reading it would conclude that the reference to "it" being "integrated into a comprehensive client/server Application," refers to the Application itself, otherwise there would be some reference to an integrated Player and Application. Moreover, two separate sets of code may be intended to work together and could

[19] Weadock Decl. ¶ 26.

[20] Weadock Decl. ¶ 26 (citing '755 patent at 7:30–36, 18:1–3).

[21] Weadock Decl. ¶ 26; '755 patent at 7:30–36.

therefore be considered "integrated," but this does not mean the two sets of code are not separate. Indeed, the reference to an "abstraction interface" in this passage suggests that the Application is separate from the Player. An interface describes the methods by which two separate and independent components communicate. This particular example is describing a situation in which the Application and Player on the device have a further requirement to communicate with a remote server, which executes "business logic." This simply means that software on the device must communicate with a central server in order to function.

39. As the rest of the paragraph points out,[22] the "abstraction interface" allows one version of the Application to run on different devices by utilizing the particular Player for that device. The discussion of "porting"[23] to other operating systems with an Application that is only implemented once further confirms that the device-specific Player and Application are separate from one another (as an Application implemented only once could not be efficiently "ported" to other operating systems if it was integrated with the device-specific code of the Player).

40. Other aspects of the specification are similar, and support Defendants' position that the Application is partitioned from the Player. For example, column 18, line 1 to line 3 (the second excerpt relied upon by Mr. Weadock), relates to two APIs—server APIs and Player APIs—that "extend the Player with Server-side or device-side Application specific logic." An API refers to an "application programming interface." The existence of two APIs relating to the Player that are used to facilitate communications with the server and Application would indicate to a POSA that the server, Player, and Application are all separate, which is why the APIs are required.

41. Mr. Weadock also asserts that the Application is not separate from the Player because the two claimed elements are "interrelated with one another."[24] But the fact that the Application and Player are interrelated has nothing to do with the fact that they are separate. For example, Mr. Weadock cites to column 5, lines 32–41 of the '755 patent, which states that the Application and Player "includes programming instructions which may be stored in memory 133 and which, when executed by processor

[22] '755 patent at 7:36–40.

[23] *Id.*

[24] Weadock Decl. ¶ 27.

135, generate the designed displays on screen 137."[25] This excerpt describes the Application and Player working in tandem, but nothing in this passage suggests that the two claimed elements are not separate sets of code, as applicant repeatedly and consistently argued during prosecution.

42. Mr. Weadock attempts to explain away the patentee's own arguments, in which the patentee emphasized the separateness of the Application and Player as one of the primary alleged distinctions between the claimed inventions and prior art to McCain.[26] Mr. Weadock opines that the applicant's reference to the advantages of "partitioning code" and the use of "two separate codes" was merely to acknowledge that "the Application is a device-independent code that is distinct from the Player."[27] Nothing in the prosecution history supports Mr. Weadock's opinion that when the applicant referred to device-independent code that is "separate" or "partitioned" from device-dependent code, the applicant really just mean "distinct." Nor does Mr. Weadock offer any explanation for what he believes makes certain code "distinct" from other code.

43. Mr. Weadock points to a portion of the prosecution history in which the applicant characterized McCain as "only providing a self-contained, device-dependent code,"[28] but that characterization is based on the applicant's view that a single code set containing device-dependent and device-independent code is "device-dependent" because there is no separate set of device-independent code. Indeed, elsewhere in the prosecution history, as discussed above, the applicant explicitly argued that McCain "clearly teaches combining all code (executable device dependent code [)] as well as parameters (device-independent code) into C2 components, which are then provided to the browser to generate a display."[29] Thus, McCain does, in fact, describe distinct code (i.e., executable code and non-executable parameters), but that was not the basis on which applicants distinguished the pending claims from McCain. Instead, the applicants differentiated their purported invention from McCain because

[25] Weadock Decl. ¶ 27; '755 patent at 5:32–41.

[26] Weadock Decl. ¶ 28.

[27] *Id.*

[28] *Id.*

[29] Exh. 4, Excerpt from 9,063,755 File History, Amendment "A" Under 37 C.F.R. §1.111, Mar. 6, 2013 at 10 (XMO_00001784 at XMO_00002803).

"[t]here is no teaching or suggestion in ***McCain*** of separating binary, non-binary, or executable components in different codes."[30]

44. A POSA reading the prosecution history would recognize that, consistent with the claims and specification, the Application must be separate from the Player, so that the Player and Application can be maintained and sent entirely separate from one another, if desired, consistent with the touted advantages of the invention. A POSA would not consider "distinct" pieces of code in the same code set to be separate from one another, because this was expressly disclosed in McCain, which applicants distinguished based on the fact that it does not disclose two sets of "different codes" as the claims require.[31]

45. ***Second***, the intrinsic evidence confirms that the Application is interpreted or executed by the Player. As an initial matter, the parties agree that the Application is "device-independent code" and, therefore, the Application cannot run on any device that does not have an appropriate program that understands and can execute or interpret the device-independent code into instructions that the device can understand. The claims provide a Player for that purpose. The Player is device-specific code that interprets or executes the Application.

46. I note that Mr. Weadock agrees that the Application and Player are interrelated with one another and goes so far as to state that the Application "depends on the Player in whole or in part to function."[32] But this relationship between the Application and Player, which I agree a POSA would recognize as a definitional requirement of both sets of code, is nowhere reflected in Express Mobile's proposed construction of Application or Player. In contrast, Defendants' proposed construction of the terms clarifies how the Application is interrelated with the Player, namely by being executed or interpreted by the Player.

[30] Exh. 4, Excerpt from 9,063,755 File History, Amendment "A" Under 37 C.F.R. §1.111, Mar. 6, 2013 at 10 (XMO_00001784 at XMO_00002803) (emphasis in original).

[31] Mr. Weadock argues that I previously acknowledged that these portions of the invention are recited separately. Weadock Decl. ¶ 28. The fact that the claims refer to the Application and the Player separately only provides additional support for Defendants' proposed construction.

[32] Weadock Decl. ¶ 27.

47. This requirement is expressly set forth in the specification, which states that the Player "interprets or executes the Application[.]"[33] The applicant made similar definitional statements in the file history.[34]

48. The operation of a device-specific Player interpreting or executing a device-independent Application is consistent with the purpose of the alleged inventions—to allow for efficient programming across multiple devices utilizing a single Application, that can be run on a variety of devices, provided each device has the "correct" Player for that particular device. Each device's Player is capable of understanding the abstractions of the Application and translating the instructions of the code into device-specific instructions that can be understood by the device.

49. Mr. Weadock suggests that the claims and specification suggest that the Application need not be interpreted or executed by the Player, but a POSA would recognize that these passages are all consistent with and support Defendants' proposed construction.

50. For example, Mr. Weadock first states that "various disclosures make it clear that the device itself can interpret or execute the Player." As written, Mr. Weadock's statement makes no sense and I believe he meant to say that the device itself can interpret or execute the ***Application***.[35] But even if this is what Mr. Weadock intended to say, neither column 6, lines 4-8 nor column 13, lines 46-49 of the '755 patent, which he cites to, support his opinion. Column 6, lines 4-8 of the'755 patent expressly confirm that the device's "Player interprets or executes the Application . . . ."

51. And column 13, lines 46-49 of the specification state that "intended programming is carried out on device 130 when the device, having the appropriate device Player, receives, and executes the device-independent Application." This statement in the specification supports Defendants'

[33] '755 patent at 6:6-7; *see also id*. at 6:9-11.

[34] See Exh. 4, Excerpt from 9,063,755 File History, Amendment "A" Under 37 C.F.R. §1.111, Mar. 6, 2013 at 9 ("the Player interprets the Application") (XMO_00001784 at XMO_00002802); *id.* at 11 ("the Player then interprets the Application") (XMO_00001784 at XMO_00002804); Exh. 7, Excerpt from 9,063,755 File History, Reconsideration after Non-Final Rejection, Jan. 16, 2014 at 9 ("the Player . . . is executed to interpret non-binary, device independent contained in the Application." [sic]) (XMO_00001784 at XMO_00002934); Exh. 8, Excerpt from 9,063,755 File History, Appeal Brief filed under C.F.R. §1.192, Nov. 26, 2014 at 8 ("the execution of the Player on the device causes the Application to be interpreted . . .") (XMO_00001784 at XMO_00002998).

[35] I reserve the right to supplement my opinion, to the extent Mr. Weadock explains this statement further in his reply report.

construction. Reading this passage, a POSA would understand that the Player on the device does execute the Application. Otherwise, there would be no need to explicitly limit the statement about executing the Application to devices "having the appropriate device Player." "Having the appropriate device Player" is a prerequisite to executing the programming on the device.

52. Mr. Weadock also states that "[d]ependent claim 21 confirms that a Player can interpret the Application, but that it must not necessarily do so."[36] I disagree. A POSA would recognize that this claim narrows independent claim 12 (from which it depends), by requiring the Player to "interpret[] dynamically received, device-independent values of the web component defined in the Application."[37] But this simply adds additional specifics about certain values in the Application that must be present and interpreted by the Player. In contrast, independent claim 12 does not require that these values be present in the Application at all. But in both claims, the Player must interpret or execute the Application, based on the portions of the specification and file history discussed above.[38]

53. Mr. Weadock also points to the portion of the specification that references a "finished page that will be displayed on screen 137 when an Application is intercepted, via a Player, on processor 135 of device 130."[39] The reference to the Application being intercepted, "via a Player, on [a] processor" would be understood by a POSA to mean that the Application is intercepted by the Player to interpret or execute it. Mr. Weadock offers no explanation for why a Player would "intercept" the Application and then take no further action on the code.

54. Column 34, lines 51-64 of '755 patent, cited by Mr. Weadock,[40] is also consistent with Defendants' proposed construction requiring the Application to be interpreted or executed by the Player. This passage discusses the Player "adapting" the Application, suggesting that the Player is executing code to modify the Application.

[36] Weadock Decl. ¶ 29.

[37] '755 patent at claim 21.

[38] If claim 21 were intended to cover the scenario where the Player interprets the Application generally, as Mr. Weadock contends, the patentee could have just claimed a Player that "interprets the Application," and there would have been no reason to specify that the Player is interpreting "dynamically received, device-independent values of the web component."

[39] Weadock Decl. ¶ 29 (citing '755 patent at 8:27-35).

[40] Weadock Decl. ¶ 29.

55. Mr. Weadock offers no support for his opinion that because the Player adapts the Application, something else must be executing the Application.

56. Indeed, Mr. Weadock agrees that the Application is device-independent code, which means that it cannot be directly executed on a specific device. Moreover, if the Application were capable of running on a device alone, it would be a device-specific solution according to the applicant. This solution would require a separate Application developed for each device, thereby defeating the stated purpose of the invention.[41] As such, a POSA would recognize that the Application must therefore be interpreted or executed by ***something*** in order to run, and that "something" is the Player. This is affirmed repeatedly throughout the specification in various embodiments:

Citation in '755 patent	**Disclosure in the Specification**	**Interpretation by Person of Ordinary Skill in the Art**
2:1–3	In another embodiment, one of the codes is an Application that is a device independent code that interpreted by the Player.	The Application is interpreted by the Player
5:22–24	Thus, in one embodiment, device 130 utilizes a Player and an Application to execute programming from authoring platform 110. A device having the correct Player is then able to interpret and be programmed according to the Application.	The Player interprets the Application
5:49–51	A Player need be provided once or updated as necessary, and thus may be used to display a large number of Applications.	The Player is used to display the Application (i.e. it interprets the Application)
5:56–64	Thus, for example and without limitation, in one embodiment, the Player transforms device-independent instructions of the Application into device-specific instructions that are executable by device 130. Thus, by way of example and without limitation, the Application may include Java programming for generating a display on screen 137, and the Player may interpret the Java and instruct processor 135 to produce the display according to the Application for execution on a specific device 130 according to the device platform.	The Player interprets the Application
6:4–8	The Application is preferably code in a device-independent format, referred to	The Player interprets or executes the Application

[41] Exh. 8, Excerpt from 9,063,755 File History, Appeal Brief filed under C.F.R. §1.192, Nov. 26, 2014 at 12 (XMO_00001784 at XMO_00002805).

	herein and without limitation as a Portable Description Language (PDL). The device's Player interprets or executes the Application to generate one or more "pages" ("Applications Pages") on a display as defined by the PDL.	
6:9–11	The Player may include code that is device-specific—that it, each device is provided with a Player that is used in the interpretation and execution of Applications.	The Player is used to interpret and execute the Application
8:31–35	Thus, for example, authoring tool 112 provides a display on screen 115 that corresponds to the finished page that will be displayed on screen 137 when an Application is intercepted, via a Player, on processor 135 of device 130.	The Player interprets the Application
9:8–10	The Player on device 130 then takes the outputs of web service 230 and binds the data to the UI components in the Application, as displayed on screen 137.	The Player interprets the Application to bind outputs of web services to that Application to display information on the screen
11:41–51	The Player determines which of the plurality of Application Pages in portion 133*b* 1 is required next. This may be determined by input actions from the Input Device 139, or from instructions from the current Application Page. The Player instructs processor 135 to extract the PDF from that Applications Page and store it in portion 133 a 1. The Player then interprets the Application Page extracted from PDL which in turn defines all of the virtual machine compliant Objects, some of which could have attributes that refer to images, audio, and/or video stored in portions 133*a*3, 133*a* 4, 133*a*5, respectively.	The Player interprets the pages contained in the Application
32:59–64	Authoring platform 110 uses compaction to transform the code and data in an intelligent way while preserving all of the original classes, methods and attributes. This requires both an intelligent server engine and client (handset) Player, both of which fully understand what the data means and how it will be used.	The Player is needed to run the Application
33:12–15	The Player, when preparing a page view for execution, decompresses and then regenerate the original objects, but this time	The Player executes the page views of the Application

	in compliance with the programming APIs of device 130.	

57. In contrast, no embodiments are described where the Application is not interpreted or executed by the Player. At the end of paragraph 29 of his report, Mr. Weadock speculates about various embodiments in which a Player "could" perform various functions that do not require interpretation or execution of the Application. But none of these embodiments are described in the specification and nothing about Defendants' proposed construction prohibits the Player from performing functions in addition to interpreting or executing the Player.

58. Based on the repeated and consistent descriptions (in both the specification and file history) of the Application being interpreted or executed by the Player, a POSA would recognize that this is part of what defines both the Application and Player. This interpretation/execution requirement is implied by the claim term "Player," and Mr. Weadock concedes that the Application "depends on" the Player to function.[42] A POSA would recognize that based on this terminology, the Player must be code that "plays" something and in this case, the specification and file history both confirm that the Player "plays" the Application, by interpreting or executing the Application.

B. Player/player

Express Mobile's Proposed Construction	Defendants' Proposed Construction
software code that facilitates the execution of an application on a device	executable device-specific code that is separate from the Application/application and that interprets or executes the Application/application

59. For the same reasons discussed above with respect to the claim term "Application"/"application," it is my opinion that the claimed "Player"/"player" is separate from, and interprets or executes, the "Application"/"application."

60. A POSA would recognize that the Player is executable code. The claim language supports the construction that the Player executes in order to carry out the claimed invention. The claim expressly indicates that the Player is "executed on the device," and each of the claims end with a limitation where the Player "provides instructions for the display of the device."

[42] Weadock Decl. ¶ 27.

61. Further, the specification states that "system 100 provides permits [*sic*] a user of authoring platform 110 to provide instructions to each of the plurality of devices 130 in the form of a device- or device-platform specific instructions for processor 135 of the device, referred to herein and without limitation as a 'Player[.]'"[43] "[I]nstructions for [the] processor" would be understood by a POSA to refer to executable code, as a processor must receive instructions in the form of an executable code in order to operate. This is further confirmed by the specification further explaining that "device-specific instructions" are those that "are executable by [the] device."[44]

62. My understanding is confirmed by the file history, where the applicant repeatedly equated "device dependent" or "device specific" code with executable code.[45] In distinguishing the prior art McCain reference, the applicant stated that "device-dependent code" is "executable."[46] The applicant reiterated this interpretation in subsequent filings during prosecution.[47] Indeed, in discussing McCain, the applicant stated that code that operates—i.e., can be executed on a device—is by definition device

[43] '755 patent 5:9–13; *see also id.* at 5:57–59 (explaining that the Player takes the device-independent code of the Application and transforms it "into device-specific instructions that are executable by [the] device"); *id.* at 5:60–64 ("[T]he Application may include Java programming for generating a display on screen 137, and the Player may interpret the Java and instruct processor 135 to produce the display according to the Application for execution on a specific device 130 according to the device platform.").

[44] '755 patent 5:57–59.

[45] Exh. 8, Excerpt from 9,063,755 File History, Appeal Brief filed under C.F.R. §1.192, Nov. 26, 2014 at 8 ("[T]he device receives an executable code in the form or a Player.") (XMO_00001784 at XMO_00002998); *id.* at 10 ("device dependent (executable) code") (XMO_00001784 at XMO_00003000); Exh. 4, Excerpt from 9,063,755 File History, Amendment "A" Under 37 C.F.R. §1.111, Mar. 6, 2013 at 10 ("executable, device-dependent code") (XMO_00001784 at XMO_00002803); *id.* at 13 ("device-dependent codes … that execute on a device") (XMO_00001784 at XMO_00002806); Exh. 7, Excerpt from 9,063,755 File History, Reconsideration after Non-Final Rejection, Jan. 16, 2014 at 9 ("Player (being device-dependent) is executed") (XMO_00001784 at XMO_00002934).

[46] *E.g.*, Exh. 4, Excerpt from 9,063,755 File History, Amendment "A" Under 37 C.F.R. §1.111, Mar. 6, 2013 at 10 ("executable, device-dependent code") (XMO_00001784 at XMO_00002803).

[47] Exh. 7, Excerpt from 9,063,755 File History, Reconsideration after Non-Final Rejection, Jan. 16, 2014 at 9 ("A C2 object includes all electronic content and functionality required to access a web service, including both device dependent (executable) and device independent (non-binary) elements that produce the selected functionality….") (XMO_00001784 at XMO_00002934); Exh. 8, Excerpt from 9,063,755 File History, Appeal Brief filed under C.F.R. §1.192, Nov. 26, 2014 at 10 (same) (XMO_00001784 at XMO_00003000).

specific and therefore a Player.[48] In characterizing the Paddon prior art reference, the applicant again stated that "device-dependent codes … execute on a device…."[49] The applicant reiterated this characterization in subsequent filings with the Patent Office during prosecution.[50] In characterizing the disclosures of the Application, the applicant placed particular emphasis on embodiments "wherein the Player (being device-dependent) is executed to interpret non-binary, device independent [code] contained in the Application."[51] The applicant explained this in detail in its appeal brief:

> As one example of the operation of the Application and Player on a device, the device receives an executable code in the form of a Player. The device also receives an Application that is a device-independent code containing web service information. The execution of the Player on the device causes the Application to be interpreted and the following actions to occur on or in communication with the device: a display requesting web service input is provided, a web service request is generated and provided to the web service, the resulting web service output is received, and a display representative of the output is generated.[52]

63. Thus, consistent with applicant's explanation, a POSA would recognize that the Player is executed to interpret the Application, which allows the device to connect to a web service to display dynamic content. The device's processor and operating system use the executable code of the Player to

[48] *E.g.*, Exh. 8, Excerpt from 9,063,755 File History, Appeal Brief filed under C.F.R. §1.192, Nov. 26, 2014 at 12 ("More specifically, since the code is generated to run on a specific device … McCain is teaching the generation of a Player.") (XMO_00001784 at XMO_00003002); *see also* Exh. 7, Excerpt from 9,063,755 File History, Reconsideration after Non-Final Rejection, Jan. 16, 2014 at 13 ("Importantly, McCain teaches that these solutions are bundled to include **all required software**, including executable code for the device…. McCain therefore teaches having a solution for each type of device. The principle of operation of McCain therefore includes utilizing solutions that are device-specific - that is, each solution [o]f McCain operates on a specific device.") (XMO_00001784 at XMO_00002938).

[49] Exh. 4, Excerpt from 9,063,755 File History, Amendment "A" Under 37 C.F.R. §1.111, Mar. 6, 2013 at 14 (XMO_00001784 at XMO_00002807).

[50] Exh. 5, Excerpt from 9,063,755 File History, Reply "B" Under 37 C.F.R. §1.116(e), May 30, 2013 at 6; Exh. 6, Excerpt from 9,063,755 File History, Response to Office Action After Final Under 37 C.F.R §1.116 and Request for Continued Examination (RCE) Under 37 C.F.R. 1.114, Sept. 26, 2013 at 12 (XMO_00001784 at XMO_00002878); Exh. 7, Excerpt from 9,063,755 File History, Reconsideration after Non-Final Rejection, Jan. 16, 2014 at 9 ("Paddon teaches device-dependent codes that execute on a device….") (XMO_00001784 at XMO_00002934); Exh. 8, Excerpt from 9,063,755 File History, Appeal Brief filed under C.F.R. §1.192, Nov. 26, 2014 at 10 (XMO_00001784 at XMO_00003000).

[51] Exh. 7, Excerpt from 9,063,755 File History, Reconsideration after Non-Final Rejection, Jan. 16, 2014 at 9 (XMO_00001784 at XMO_00002934).

[52] Exh. 8, Excerpt from 9,063,755 File History, Appeal Brief filed under C.F.R. §1.192, Nov. 26, 2014 at 8 (XMO_00001784 at XMO_00002998).

run the Player, where the Player in turn executes ('plays') the Application's instructions. Neither Express Mobile nor Mr. Weadock has identified any intrinsic evidence to the contrary.

64. Mr. Weadock refers to the specification stating that "the Player can 'adapt[] the Application to the resources and limitations of any particular device'" (Weadock Decl. ¶ 29 (citing '755 patent 34:51–64)), and I understand Express Mobile argues that this passage indicates that the Player need not be executable code (Opening Brief at 7). I disagree. This passage describes a Player that adapts the Application to the device based on the hardware limitations of the device. During this adapting, the Player will need to be executing code to carry out the adapting functions.

65. I also understand that Express Mobile asserts that the reference to the Player "extend[ing] the operating system" supports its position that the Player is not an executable. (Opening Brief at 7.) This is incorrect. An operating system is an executable, so a Player that extends the operating system is also an executable.

66. Additionally, it is my opinion that a POSA reading the specification and file history would understand the claimed "Player"/"player" to be device-dependent, or device-specific code.[53] The specification states that instructions "in the form of a device- or device-platform specific instructions for [a] processor of the device" is "***referred to herein and without limitation*** as a 'Player[.]'"[54] This is confirmed by the file history, where the applicant stated: "The present patent application ***consistently***

[53] I use these terms interchangeably herein.

[54] '755 patent at 5:8–24; *see also* Exh. 8, Excerpt from 9,063,755 File History, Appeal Brief filed under C.F.R. §1.192, Nov. 26, 2014 at 6 (applicant referencing paragraph [0048], i.e., 5:8-24 as issued, as supporting the disclosure of a Player) (XMO_00001784 at XMO_00002996).

use [sic] the words 'Application' (with a capital A) and 'Player' (with a capital P) to refer to code that is provided to devices for accessing web services, where . . . a Player is **device-dependent code**."[55]

67. Mr. Weadock argues that a POSA would conclude that "players within the '755 family may be either device-dependent or device-independent."[56] I disagree. The primary support for Mr. Weadock's opinion is a passage from the specification that states:

> In one embodiment, one of the codes is a Player, which is a thin client architecture that operates in a language that manages resources efficiently, is extensible, supports a robust application model, and has no device specific dependencies.[57]

68. Mr. Weadock interprets this passage to be referring to a Player with "no device specific dependencies," but in my opinion, a POSA would recognize that this "no device specific dependencies" clause refers to the "language," not the "Player."[58] Specifically, the sentence is more reasonably read as "In one embodiment, one of the codes is a Player, which is a thin client architecture that operates in a language that **[1]** manages resources efficiently, **[2]** is extensible, **[3]** supports a robust application model, and **[4]** has no device specific dependencies."[59] That is, the numbered items modify "language," not "Player." I believe this interpretation is much more consistent with the specification, and in fact, is

[55] Exh. 8, Excerpt from 9,063,755 File History, Appeal Brief filed under C.F.R. §1.192, Nov. 26, 2014 at 9–10 n.2 (XMO_00001784 at XMO_00002999-3000) (first emphasis added). *See also* Exh. 4, Excerpt from 9,063,755 File History, Amendment "A" Under 37 C.F.R. §1.111, Mar. 6, 2013 at 11 ("device dependent code (such as the claimed Player)") (XMO_00001784 at XMO_00002804); *id.* at 12 ("device-dependent code (the Player)") (XMO_00001784 at XMO_00002805); *see also* Exh. 7, Excerpt from 9,063,755 File History, Reconsideration after Non-Final Rejection, Jan. 16, 2014 at 11 (distinguishing prior art as it "teaches only generating device independent code," and providing no teaching or suggestion for "generating or providing device-dependent code (a Player).") (XMO_00001784 at XMO_00002936); Exh. 8, Excerpt from 9,063,755 File History, Appeal Brief filed under C.F.R. §1.192, Nov. 26, 2014 at 8 ("If a new device comes on the market, or if it is found that the device-dependent code for a specific device needs an update, a new Player is developed and provided to those specific devices.") (XMO_00001784 at XMO_00002998); *id.* at 12 (claiming that the prior art taught the generation of a Player "since the code is generated to run on a specific device") (XMO_00001784 at XMO_00003002).

[56] Weadock Decl. ¶ 32.

[57] Weadock Decl. ¶ 30; '755 patent at 1:59–62.

[58] It is also possible that the inclusion of the word "no" is a typographical error, of which there are many in the '755 patent. *See, e.g.*, '755 patent at claim 12, erroneously claiming an Application that is "device-dependent code."

[59] '755 patent at 1:59–62 (annotated).

supported by the discussion at column 6, lines 48-56 of the Portable Description Language ("PDL") being a common language for the authoring tool, Application, and Player.

69. This interpretation also better conforms to how a POSA would understand the sentence, because the four criteria listed would more easily be understood as characteristics of a language rather than of an architecture. A language utilizes resources in the form of how it represents program variables in memory, and languages are often extensible. It is reasonable to consider what a language is capable of modelling, given its syntax and semantics. And most to the point, most high-level languages have no device specific dependencies at all, but rather rely on the operating system utilities or libraries which support the language to worry about such details. For example, the Java language statement System.out.println ("Hello World!"); writes "Hello World!" on the screen. All the details of how to do that is handled in a system utility that the programmer need know nothing further about. Mr. Weadock suggests that a POSA would not use the term "extensible" to describe a programming language, or state that a language "supports a robust application model," but I disagree. A POSA would readily understand that an extensible language refers to a language that allows users to modify or add syntax.[60] And a language that supports a robust application model simply means that the language is well suited for writing applications.

70. Moreover, my view of how a POSA would interpret this phrase is supported by the prosecution history, which includes a provisional application in the form of an Express Mobile User's Guide. That document provides that "[a] thin client architecture requires that a language be adopted that manages resources efficiently, is extensible, supports a robust application model, and has no device specific dependencies."[61] This language closely tracks the portion of the specification relied upon by Mr. Weadock and would further confirm for a POSA that "no device specific dependencies" refers to the language, not the Player.[62]

[60] *See* Exh. 10, Daniel Zingaro, *Modern Extensible Languages* (Oct. 2007), http://www.cas.mcmaster.ca/sqrl/papers/SQRLreport47.pdf, at Abstract.

[61] Exh. 3, Excerpt from 9,063,755 File History, Express Mobile User Guide Appendix D, "Systems and Methods for Presenting Information on Mobile Devices" at 21 (XMO_00001784 at XMO_00002316).

[62] The quoted language from the Express Mobile User Guide also refutes Mr. Weadock's view that a POSA would not describe a language as "extensible" or supportive of "a robust application model," since the inventors themselves described languages as such.

71. In contrast, if one were to accept Mr. Weadock's interpretation of the passage as describing a thin client *architecture*, the clause about no device dependencies makes no sense at all, because a "thin client" is all about device dependence. A thin client architecture is one in which almost everything is done in a server on the network, and the client itself just worries about outputs (drawing on the screen) and inputs (key presses, touches, mouse, etc.); this is why the client is referred to as "thin." To a POSA, the concept of a thin client with no device specific dependencies would be nonsensical.

72. Mr. Weadock also includes a list of citations to various portions of the specification,[63] but it is unclear why Mr. Weadock includes these citations, because he does not discuss any of them in detail, and an examination of the statements in fact confirms applicant's representation to the PTO that "a Player is device-dependent code."[64] The table below illustrates how none of the citations provided by Mr. Weadock support a different reading:

Citation in '755 patent	Description of the Disclosure in the Specification	Interpretation by Person of Ordinary Skill in the Art
3:58–62	This section refers to device-specific routines, "that is, codes that are specific to the operating system, programming language, or platform of specific devices." The citation continues after the section cited by Mr. Weadock to state that the invention is not limited to using Java-based systems, but that "one skilled in the art could provide Players for devices using routines provided	Supports idea that Player is device specific / device dependent

[63] Weadock Declaration ¶ 32.

[64] Exh. 8, Excerpt from 9,063,755 File History, Appeal Brief filed under C.F.R. §1.192, Nov. 26, 2014 at 9-10 n.2 (XMO_00001784 at XMO_00002999) (first emphasis added). *See also* Exh. 4, Excerpt from 9,063,755 File History, Amendment "A" Under 37 C.F.R. §1.111, Mar. 6, 2013 at 11 ("device dependent code (such as the claimed Player)") (XMO_00001784 at XMO_00002804); *id.* at 12 ("device-dependent code (the Player)") (XMO_00001784 at XMO_00002805); *see also* Exh. 7, Excerpt from 9,063,755 File History, Reconsideration after Non-Final Rejection, Jan. 16, 2014 at 11 (distinguishing prior art as it "teaches only generating device independent code," and providing no teaching or suggestion for "generating or providing device-dependent code (a Player).") (XMO_00001784 at XMO_00002936); Exh. 8, Excerpt from 9,063,755 File History, Appeal Brief filed under C.F.R. § 1.192, Nov. 26, 2014 at 8 ("If a new device comes on the market, or if it is found that the device-dependent code for a specific device needs an update, a new Player is developed and provided to those specific devices.") (XMO_00001784 at XMO_00002998); *id.* at 12 (claiming that the prior art taught the generation of a Player "since the code is generated to run on a specific device") (XMO_00001784 at XMO_00003002).

Citation in ’755 patent	Description of the Disclosure in the Specification	Interpretation by Person of Ordinary Skill in the Art
	on a platform.” ’755 patent at 3:62–67.	
5:8–24	This citation is to the statement that “system 100 provides permits [sic] a user of authoring platform 110 to provide instructions to each of the plurality of devices 130 in the form of a device- or device-platform specific instructions for processor 135 of the device, referred to herein and without limitation as a “Player[.]”	Supports idea that Player is device specific / device dependent
5:42–55	This citation states that the authoring tool may “produce and store within memory 111*a* a plurality of Players (for different devices 130).” It continues: “A Player need be provided once or updated as necessary, and thus may be used to display a large number of Applications.” It goes on to state that “This is advantageous for the authoring process, since all of the device-dependent programming is provided to a device only once (or possibly for some small number of upgrades)[.]”	Supports idea that Player is device specific / device dependent
5:56–6:3	This citation states that “in one embodiment, the Player transforms device-independent instructions of the Application into device-specific instructions that are executable by device 130.”	Supports idea that Player is device specific / device dependent
6:9–17	This citation states that “The Player may include code that is device-specific—that it, each device is provided with a Player that is used in the interpretation and execution of Applications.”	Supports idea that Player is device specific / device dependent
7:13–20	This citation refers to a PDL, which is discussed in the context of the Application, and not the Player. *See* ’755 patent at 7:1–2	Citation is irrelevant to question of whether Player is device specific / device dependent

Citation in '755 patent	Description of the Disclosure in the Specification	Interpretation by Person of Ordinary Skill in the Art
	("The terminology used here is a PDL, that is the 'internal database' of Rempell is equivalent to the PDL of the present Application."); *id.* at 6:4–6 ("The Application is preferably code in a device-independent format, referred to herein and without limitation as a Portable Description Language (PDL)."); *id.* at 6:48–49 ("In certain embodiments, the applicant model used in developing and providing Applications is a PDL."). An ordinary artisan would understand the discussion of this citation to be referring to the Application.	
7:30–40	This citation is referring to how the Player is used to "separate[] all device, operating system and virtual machine dependencies from the Player's Application model business logic[.]" An ordinary artisan would understand the discussion of this citation to be referring to the Player including device, operating system, and virtual machine dependencies, i.e. that the Player is device-dependent, and an "abstraction interface" of the Player is used to allow the Application and server to interact with the device.	Supports idea that Player is device specific / device dependent
8:7–17	This citation is referring to how a "response director" determines the appropriate "Player (each usable by ***certain*** devices)". It also states that the device provides "device-specific information" in order to allow the "response director" to "determine[] the appropriate Player for the device." An	Supports idea that Player is device specific / device dependent

Citation in ’755 patent	Description of the Disclosure in the Specification	Interpretation by Person of Ordinary Skill in the Art
	ordinary artisan would understand this to be disclosing that device-specific information needs to be provided to the response director because Players are device-specific.	
8:27–35	This citation states how the authoring tool can produce both an Application and a Player, and how the Application is intercepted, via a Player, in order to show a screen. An ordinary artisan would understand it to be referring to how the authoring tool produces the Player and the Player intercepts the Application, not whether the Player is device-dependent or not.	Citation is irrelevant to question of whether Player is device specific / device dependent
9:4–10	This citation states how the Player receives outputs of web services and binds the data to the UI components in the Application. An ordinary artisan would understand this statement to be referring to how the authoring tool produces the Player and the Player performs various functions. This is consistent with the notion that the Player is device-specific code, as performance of functions is accomplished through device-specific code for a processor.	Supports idea that Player is device specific / device dependent
11:41–51	This citation states how the Player instructs a processor to carry out some acts. As discussed above, instructions to a processor are by definition device-dependent codes.	Supports idea that Player is device specific / device dependent
17:66–18:3	This citation states how there can exist a “Universal Player” that includes “all the code libraries for the Player.” An ordinary artisan would	Citation is irrelevant to question of whether Player is device specific / device dependent

Citation in ’755 patent	Description of the Disclosure in the Specification	Interpretation by Person of Ordinary Skill in the Art
	understand this statement to be referring to how Players can have different libraries, not whether a Player is device-dependent or not.	
23:43–46	This citation states that the authoring tool generates a Player “specific to device 130 of Fig. 4B.”	Supports idea that Player is device specific / device dependent
33:12–15	This citation states that the Player decompresses objects “in compliance with the programming APIs of device 130.” An ordinary artisan would understand that this paragraph describes various functions the Player performs. This is consistent with the notion that the Player is device-specific code, as performance of functions is accomplished through device-specific code for a processor.	Supports idea that Player is device specific / device dependent
33:26–28	This citation discusses how the response director “provide[s] the correct Player to a given device based on the information the device sent to it,” similar to the statement at 8:7–17. An ordinary artisan would understand this to be disclosing that device-specific information needs to be provided to the response director because Players are device-specific.	Supports idea that Player is device specific / device dependent
Fig. 2A	Fig. 2A is described as “a schematic of a system 200 of an embodiment of system 100 illustrating the communications between different system components.” ’755 patent at 7:63–65. An ordinary artisan would understand this figure to show how the authoring platform sends all Players to the response director, and the	Supports idea that Player is device specific / device dependent

Citation in '755 patent	Description of the Disclosure in the Specification	Interpretation by Person of Ordinary Skill in the Art
	response director provides a device-specific Player to the device. *See id.* at 8:7–17 (discussing the role of the response director shown in Fig. 2A).	
Fig. 2B	Fig. 2B shows the various software components that require space in and access to the heap in memory. The term "heap" relates to memory management. The boxes show various aspects that need space in the heap. *See* '755 patent at 10:62–11:2. In addition, the connections are described as "logical connection[s] between the different types of programming stored in [the] heap." *Id.* at 11:36–40. An ordinary artisan would understand Figure 2B to relate to memory management, and not whether the Player is device-dependent or not.	Citation is irrelevant to question of whether Player is device specific / device dependent

73. Mr. Weadock also contends that the discussion of a "thin client architecture" elsewhere in the '755 patent supports his view that a Player need not be device-dependent code.[65] I disagree. In my opinion, Mr. Weadock is confused as to the meaning of "thin client" and why the ***architecture*** (not the "thin client" itself) is referred to as "device independent."[66] Specifically, as one dictionary defines it, "in a client/server architecture," a thin client is "a client computer that performs little or no data processing. The processing is instead performed by the server."[67] A "thin client" can also be considered to be "a

[65] *See* Weadock Decl. ¶ 33.

[66] *See* Weadock Decl. ¶ 33.

[67] *See* Exh. 11, Microsoft Computer Dictionary (3d Ed.) at 466 (defining "thin client") (DEF-WDTX0773 at DEF-WDTX0776).

network device that has the ability to process information independently but relies on servers for applications, data storage, and administration."[68]

74. Importantly, "client/server architecture" has its own technical meaning. As stated in Microsoft's Computer Dictionary (4th Ed.) that Mr. Weadock relies on, a client/server architecture is:

> An arrangement used on LANs (local area networks) that makes use of distributed intelligence to treat both the server and the individual workstations as intelligent, programmable devices, thus exploiting the full computing power of each. This is done by splitting the processing of an application between two distinct components: a 'front end' client and a 'back-end' server. The client component is a complete, stand-alone personal computer (not a 'dumb' terminal), and it offers the user its full range of power and features for running applications. . . . The client and server machines work together to accomplish the processing of the application being used. Not only does this increase the processing power available over older architectures but it also uses that power more efficiently. The client portion of the application is typically optimized for user interaction, whereas the server portion provides the centralized, multiuser functionality.[69]

These definitions help explain what is meant by "device independence" in the context of thin clients working on a client/server architecture. Specifically, in a thin client architecture, the clients rely on the server to provide data processing. The various clients must have a way to "talk" to the server. This is accomplished by having the server provide a device-independent, known API for clients to use in order to access the processing power of the server. In the words of the '755 patent, this is a "language" that "has no device specific dependencies."[70]

75. The only passage in the specification of the '755 patent cited by Mr. Weadock is consistent with my understanding of thin client architectures.[71] The passage at column 6, lines 22-30

[68] *See* Opening Brief Exh. 24, XMO-LIT00054035 at XMO-LIT00054039 (Microsoft Computer Dictionary (4th Ed.)) at 441 (defining "thin client"). To be clear, the ability to process information independently in this definition is referring to processing information independently from any server or other computer, ***not*** as Mr. Weadock suggests, that the thin client is itself "device-independent," which has its own meaning in the art.

[69] *See* 12, Microsoft Computer Dictionary (4th Ed.) at 88 (definition of "client/server architecture") (DEF-WDTX0777 at DEF-WDTX0779).

[70] '755 patent at 1:60-62. This is also consistent with the definition of "device-independence" as a "characteristic of a program, ***interface, or protocol*** that supports software operations that produce similar results on a wide variety of hardware." *See* Exh. 13, Microsoft Computer Dictionary (3d Ed.) at 142 (DEF-WDTX0773 at DEF-WDTX0775); Exh. 12, Microsoft Computer Dictionary (4th Ed.) at 135 (same) (DEF-WDTX0777 at DEF-WDTX0780).

[71] Weadock Decl. ¶ 33 (citing '755 patent at 6:22-30).

describes the Player being a "thin client" in that the Player "depends primarily on a central server for processing activities, and mainly focuses on conveying input and output between the user and the server."[72]

76. But this characterization of the Player as a "thin client" does not support Mr. Weadock's opinion that the Player is not device-specific code. There must be code to convey input and output between the user and the server. Mr. Weadock argues in thin client architectures "device-specific code would not be needed for the player except for the processing of those generic I/O operations."[73] But Mr. Weadock's statement proves the point: even in thin client architectures, device-specific code is necessarily to process I/O operations. In the context of the '755 patent, the specification describes this thin client as focusing on "conveying input and output between the user and the server."[74] This is processing I/O operations. ("I/O" refers to input/output). There is no reason why, merely because the Player is part of a thin client architecture, that it cannot be device-dependent. A "thin" client is referred to as such, because it has little logic of its own, instead sending the user inputs on to a server for processing, and drawing on the display what the server requests of it. But no POSA would consider a "thin client" to imply complete device independence as Mr. Weadock suggests and none of the exhibits he points to suggest otherwise.[75] Instead, a POSA would understand a thin client to be just the opposite; it contains *all* the device dependencies, and all the device independent processing takes place on the server communicating with the thin client.

77. Finally, Mr. Weadock also claims that the issue as to whether Player is "device-dependent" only pertains to the '044 patent, since the claims of the '755 and '287 patents explicitly require the Player to be device-dependent code, whereas the claims of the '044 patent do not. [76] Thus,

[72] '755 patent at 6:24-30.

[73] Weadock Decl. ¶ 33.

[74] '755 patent at 6:29–30.

[75] Weadock Decl. ¶ 58 (citing Kramer Decl. Ex. 6M, 6N, 6O, 6P). I note that Kramer Decl. Exhibit 6O is an IBM patent, which does discuss the use of "device independent" data, but it also describes "device dependent data streams" in thin client architectures, which further supports my opinion that Mr. Weadock's emphasis on thin client architectures has nothing to do with whether the claimed Player is device-dependent code.

[76] Weadock Decl. ¶ 34; '755 patent at 37:29–30; '287 patent at 38:19–20; '044 patent at claim 1.

Mr. Weadock claims that, to the extent any "device-dependent" requirement is necessary, it would only apply to the '044 patent because "there would be no need to include this limitation in some patent claims and not others if the Player was so clearly limited to device dependent code."[77]

78. Mr. Weadock has provided claim construction opinions regarding the term "Player/player" in four litigations involving the '755, '287, and '044 patents, and this is the first time he has made this assertion. Indeed, in the *Shopify* litigation, Mr. Weadock testified that he was not advocating for a different construction of "Player" with a capital P as in the '755 and '287 patents, compared to "player" with a lowercase p as used in the '044 patent.[78]

79. Furthermore, a POSA would recognize that the '044 patent shares a common specification with the '755 and '287 patents, which states that instructions "in the form of a device- or device-platform specific instructions for [a] processor of the device" is "referred to herein and without limitation as a 'Player[.]'"[79] The applicant repeatedly argued that the Player is device-specific code and, during prosecution of the '044 patent, stated that the claims are "patentable for at least the reasons as those patents [the '755 patent]."[80] While the claim language in the '044 patent does differ in some regard for the language in the '287 and '755 patents, none of these differences would suggest to a POSA that the Player could be device-independent ***or*** device-dependent code as Mr. Weadock now contends.

80. In addition, a POSA would not interpret the prosecution history upon which Mr. Weadock relies to suggest that Player in the '044 patent is not device-dependent.[81] The passage focuses on the capability of the authoring tool to "store information representative of said defined UI object and related settings in a database;" and "retrieve said information," wherein the Application and Player support this functionality.[82] This language says nothing as to whether the claimed "player" in the '044 patent claims

[77] Weadock Decl. ¶ 34.

[78] Exh. 19, March 25, 2020 Deposition of Glenn E. Weadock, *Shopify Inc., et al v. Express Mobile, Inc.*, Case No. 19-439 (D. Del.) at 189:25–3 ("Q. You're not advocating for any difference in construction of those terms depending on whether a capitalization is used or not, are you? A. No. I'm offering no opinion that.").

[79] '044 patent at 5:14–18.

[80] Exh. 9, Excerpt from 9,928,044 File History, Preliminary Amendment at Remarks, Sept. 17, 2017.

[81] Weadock Decl. ¶ 36.

[82] Weadock Decl. ¶ 36.

is device-dependent code. Other portions of the specification and prosecution history for the ’755 patent repeatedly define the Player as device-dependent code and nothing would suggest to a POSA that the applicant intended the Player to be device-dependent code in some claims, but not in others.

81. Mr. Weadock further states that the ’044 patent relates to a different embodiment—specifically an embodiment of a Player that “utilizes information from said database to generate for the display of at least a portion of said one or more web pages”—but he points to no disclosures that identify this “different embodiment.”[83] This is because every embodiment described in the specification involves a device-specific Player. The notion that a Player utilizes information from a database does not suggest that the claimed Player is not device-specific code.

C. device-dependent code / device dependent code

Express Mobile’s Proposed Construction	Defendants’ Proposed Construction
code that is specific to the operating system, programming language, or platform of a device	code for a specific device

82. In my opinion, Defendants’ proposed construction of “device-dependent code” or “device dependent code” more accurately reflects how a POSA would understand the terms at the time of the invention. The specification of the ’755 patent discloses that “different devices 130 may be operable using different sets of instructions, that is having one of a variety of different ‘device platforms.’”[84] In the paragraph immediately following this statement is the definition of “Player,” defined to be a “device- or device-platform specific instructions for processor 135 of the device.”[85]

83. Elsewhere, the specification states that:

> Routines 114 may include device-specific routines—that is, codes that are specific to the operating system, programming language, or platform of specific devices 130, and may include, but are not limited to Java, Windows Mobile, Brew, Symbian OS, or Open Handset Alliance (OHA).[86]

84. The specification also explains that the authoring tool generates code that is “parsed into instructions used by different device platforms and instructions that are independent of device

[83] Weadock Decl. ¶ 35.

[84] ’755 patent at 4:66–5:1.

[85] *Id.* at 5:9-12.

[86] ’755 patent at 3:58-62.

platform."[87] The specification explains that splitting the code into a device-independent Application and a device-dependent Player is "advantageous for the authoring process, since all of the device-dependent programming is provided to a device only once . . . permitting a smaller Application, which is the same for each device."[88]

85. And the specification discusses sending a "correct" or "appropriate" Player to a device, based on, e.g., what operating system is on the device.[89] During prosecution, the applicant distinguished prior art that allegedly included "device-specific" code, in arguing that "device-specific" meant "each solution [of the prior art] operates on a specific device."[90]

86. Defendants' proposed construction of "device-dependent code" is also supported by definitions from technical dictionaries. For example, Microsoft's Computer Dictionary (3d Ed.) defines device dependence as "the requirement that a particular device be present or available for the use of a program, interface, or protocol. Device dependence in a program is often considered unfortunate because the program either is limited to one system or requires adjustments for every other type of system on which it is to run."[91]

87. Similarly, the IBM Dictionary of Computing defines "device-dependent" as "(1) pertaining to a function that can be accomplished only if particular types of devices are available.

[87] *Id.* at 5:18–20.

[88] *Id.* at 5:51–55.

[89] *See* '755 patent at 33:26–28 ("Response Director operates on a network connected computer to provide the correct Player to a given device based on the information the device sent to it."); *id.* at 33:44–46 ("User agent database 1201 includes user agent information regarding individual devices 130 that are used to identify the operating system on the device."); *id.* at 33:57–61 ("The User Agent prompts a database lookup in user agent database 1201 which returns data including, but not limited to, make, model, attributes, MIDP 1.0 MIDP 2.0, WAP and distinguishes the same models from different countries."); *id.* at 34:46–50 ("As an example of system 1300, when response director 210 receives an SMS message from device 130, the response director identifies the device characteristics operator and locale from database 1301 and a Player URL from database 1303 and provides the appropriate Player to the device.").

[90] Exh. 7, Excerpt from 9,063,755 File History, Reconsideration after Non-Final Rejection, Jan. 16, 2014 at 13 (XMO_00001784 at XMO_00002938); *see also* Exh. 8, Excerpt from 9,063,755 File History, Appeal Brief filed under C.F.R. §1.192, Nov. 26, 2014 at 8 ("If a new device comes on the market, or if it is found that the device-dependent code for a specific device needs an update, a new Player is developed and provided to those specific devices.") (XMO_00001784 at XMO_00002998).

[91] Exh. 13, Microsoft Computer Dictionary (3d Ed) at 142 (DEF-WDTX0773 at DEF-WDTX0775); *see also* Exh. 12, Microsoft Computer Dictionary (4th Ed) at 135 (same) (DEF-WDTX0777 at DEF-WDTX0780).

(2) pertaining to a program that can be executed successfully only if particular types of devices are available. (3) Contrast with device-independent."[92] Code that only works "if particular types of devices are available" is device-dependent, consistent with Defendants' proposed construction, because it is specific for and will only run on those particular types of devices.

88. I disagree with Mr. Weadock that all that is required is that the code is "aware of device type or device-specific instructions."[93] First, Mr. Weadock quotes the specification at 34:4-11.[94] But this paragraph is simply a discussion of how the process of installing a Player on a device via a response director can include sending configuration information in a text file. A person of ordinary skill in the art would not understand this discussion to relate to the meaning of "device dependent code" given the definition of Player and discussion of device-dependent and device-specific code elsewhere in the specification and file history.

89. Mr. Weadock also cites to column 34, lines 51-64 of the '755 patent in support of Express Mobile's construction.[95] This section of the specification again discusses how the Player may change its operation based on the characteristics of a particular device. But this is consistent with the fact that the Player is "device- or device-platform specific instructions for processor 135 of the device."[96] Instructions for a processor are executable code that differs depending on the device platform. Mr. Weadock also cites to column 5, lines 56-64 of the '755 patent, but this section of the specification again supports Defendants' construction as it references the Player instructing a processor.

[92] Exh. 14, IBM Dictionary of Computing (10th Ed) at 193 (DEFSEXTRINSIC_XMO_00000053).

[93] Weadock Decl. ¶ 39.

[94] Mr. Weadock, in his description of this passage at 34:4-11 of the '755 patent, also refers to the "Player" as "containing 'device-dependent code.'" Weadock Decl. ¶ 39. Notably, however, this passage of the '755 patent doesn't describe the Player has having device-dependent code or the relationship between the terms "player" and "device-dependent code." *See* '755 patent at 34:4–11. Elsewhere, Mr. Weadock argues that the Player is ***not*** necessarily device-dependent code. *See* Weadock Decl. ¶ 32 (arguing that Players need not be device specific, and "Players within the '755 patent family may be either device-dependent or device-independent."). Thus, it is unclear, under Mr. Weadock's understanding, why he is relating this passage to "device-dependent code" when he believes nothing about a Player, absent more, implies that it must be device-dependent. To be clear, as described above, I believe a Player must be device-dependent code.

[95] Weadock Decl. ¶ 39.

[96] '755 patent at 5:10–12.

90. In general, it is my opinion that Mr. Weadock is confusing the idea of configuration, where code can take advantage of particular hardware or software characteristics (e.g., screen size), with device-dependence, where executable code is written for the specific device's processor or operating system and cannot be run on devices or platforms with different processors (e.g. an Intel processor vs. an ARM processor) or operating systems (e.g. Windows vs. MacOS). "[A]wareness of the platform" as proposed by Express Mobile, is not the ordinary meaning of the term "device dependence," and portions of the specification cited by Mr. Weadock do not define what device-dependent code is. These citations merely describe how a Player can be configured.

91. I understand that Express Mobile's proposed construction is based on the Court's construction in the *Shopify* and *GoDaddy* cases and is based on language in the specification referring to device "platforms," as discussed above. My primary objection to this construction is the use of the word "platform," because subsequent to the Court's claim constructions in those cases, it has become apparent that Express Mobile is twisting the word "platform" in the Court's construction to improperly capture any code that differs based on device configuration, regardless of whether the code is actually specific for any device.

92. For example, in the *Shopify* litigation, Express Mobile argues that conditional JavaScript, which is sent to every device, is nonetheless "device-dependent code," because different portions of the code may be executed depending on the version of a web browser that a client's computer is running. A POSA would not consider this type of code to be "device-dependent" as that term is used in the claims of the '755, '287, and '044 patents, because it is provided to, and runs on the range of supported devices. The fact that some code may not actually be executed by certain web browsers does not make the code that is executed device dependent.

93. Indeed, this type of conditional logic, which is provided to, and runs on a range of devices, is nowhere referenced in the specification as "device-dependent" code. A POSA would have been aware of such code as of 2008, because it had been in existence for many years at that point in time.[97] But the complete absence of discussion of any such code in the specification would have

[97] Exh. 20, Nick Heinle, Designing with Javascript-Creating Dynamic Web Pages (1997).

conveyed to a POSA that the inventors did not consider such code part of the claimed "device-dependent" code. Instead, consistent with the specification and the plain and ordinary meaning of device-dependence, a POSA would recognize that device-dependent code means code for a specific device. I disagree with Mr. Weadock's assertion that code for a "Dell Latitude D830 running Windows 95 version OSR" was not what the inventor intended when referring to "device-dependent code." Windows is an operating system and thus, code specific to Windows machines would have been considered device-dependent code by a POSA in view of the specification and file histories.

D. device-independent code / device independent code

Express Mobile's Proposed Construction	Defendants' Proposed Construction
No construction necessary ***Alternative:*** code that is not specific to the operating system, programming language, or platform of a device.	code that is not for a specific device

94. Similarly, it is my opinion that a POSA would construe "device-independent code" or "device independent code" to mean code that is not for a specific device.

95. Defendants' proposed construction of "device-independent code" aligns with its proposed construction of "device-dependent code," and is consistent with the specification, which as discussed above, describes the purported invention of including code that is what has historically been called "write-once-run-anywhere."[98] This concept is consistent with what the '755, '287, and '044 patents describe. For example, the specification explains that splitting the code into a device-independent Application and a device-dependent Player is "advantageous for the authoring process, since all of the device-dependent programming is provided to a device only once . . . permitting a smaller Application, which is ***the same for each device***."[99] A POSA would understand this reference to code that is the same for each device, means that the code is not specific to any particular device.

96. My opinion is again supported by the IBM Dictionary of Computing, which defines device-independent "(1) pertaining to a function that can be accomplished without regard for the

[98] *See* Exh. 21, "Write once, run anywhere?", ComputerWeekly.com (May 2, 2002), https://www.computerweekly.com/feature/Write-once-run-anywhere.

[99] '755 patent at 5:51–55 (emphasis added).

characteristics of particular types of devices. (2) pertaining to a program that can be executed successfully without regard for the characteristics of particular types of devices. . . (3) Contrast with device-dependent."[100]

E. web component

Express Mobile's Proposed Construction	Defendants' Proposed Construction
one or more functionalities associated with one or more web page elements to be displayed on a device	software object that provides functionalities of a web service

97. In my opinion, Defendants' proposed construction accurately reflects how a POSA would understand the term "web component."

98. The term "component" is a term of art in computer science and web design fields, and I agree with Defendants' construction, which is consistent with the definition set forth in the W3C glossary,[101] which the patentee relied on during prosecution to explain the underlying technology to the examiner.[102]

99. The first W3C definition for "component" is:

> A component is a software object, meant to interact with other components, encapsulating certain functionality or a set of functionalities. A component has a clearly defined interface and conforms to a prescribed behavior common to all components within an architecture.[103]

Defendants' proposed construction reflects the concept of "functionality" in this definition and is consistent with how a POSA would understand the term.

100. The specification further confirms that what makes certain software objects the claimed "web components," is the fact that they are objects that provide functionalities associated of web service.

[100] Exh. 14, IBM Dictionary of Computer (10th ed.) at 193 (defining "device-independent") (DEFSEXTRINSIC_XMO_00000053).

[101] *See* Exh. 22, W3C Working Group Note 11 February 2004, https://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/.

[102] *See* Exh. 5, Excerpt from 9,063,755 File History, Reply "B" Under 37 C.F.R. §1.116(e), May 30, 2013 at 5 (XMO_00002852); Exh. 6, Excerpt from 9,063,755 File History, Response to Office Action After Final Under 37 C.F.R §1.116 and Request for Continued Examination (RCE) Under 37 C.F.R. 1.114, Sept. 26, 2013 at 11 (XMO_00002877).

[103] Exh. 22, W3C Working Group Note 11 February 2004, https://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/.

For example, the specification discusses how "[e]ach web service is identified and/or defined as an entry in web component registry 230 . . . . Web component registry 230 is provided through server 120 to authoring platform 110 so that a user of the authoring platform may bind web services 230 to elements to be displayed on the device 130[.]"[104] The specification also discusses how "[a]uthoring platform 110 further permits a user of the authoring platform to associate objects, such as objects for presenting on screen 137, with components of one or more web services 230 that are registered in web component registry 220. In one embodiment, information is provided in an XML file to web component registry 220 for each registered components of each web service 230. Web component registry 220 may contain consumer inputs related to each web service 230, environmental data such as PIM, time or location values, persistent variable data, outputs related to the web service, and/or optional hinting for improving the user's productivity."[105] Additionally, the specification discloses that "[i]n one embodiment, a component of web service 230 is selected from authoring platform 110 which presents the user with WYSIWYG dialog boxes that enable the binding of all the inputs and outputs of component of web service 230 to a GUI component of the Application as will be displayed on screen 137."[106]

101. Figures 3E and 3F further illustrate the fact that web components are objects that provide functionalities to a web service and depict "a pop-up menu for adding web components."[107] The specification also describes utilizing web components to build an Application. It teaches that web components may be "added, edited, or removed from a selected object."[108] Menus are provided to "select a Web Component" and "present[] a list of web components . . . obtained from web component registry 220."[109]

102. The file history, which includes the priority documents further describing web components, provides additional support for Defendants' proposed construction. For example, one of

[104] '755 patent at 8:18–26.

[105] *Id.* at 8:36-47.

[106] *Id.* at 8:58–63.

[107] *Id.* at 2:33-34.

[108] *Id.* at 22:15-16.

[109] *Id.* at 22:23-29.

these documents describes an Appendix of "Web Component Models."[110] These exemplary models are XML formatted documents that appear to define inputs and outputs for various web services.[111] They encapsulate certain functionality or a set of functionalities (e.g. functionalities relating to web services).

103. Express Mobile's construction recites the functional nature of web components, but it fails to set forth the relationship between web components and web services and conflates "web services" with "web pages." Mr. Weadock acknowledges that "[w]eb components comprise functionalities associated with, for example, web services,"[112] but this is not set forth in Express Mobile's proposed construction. Instead, Express Mobile's proposed construction incorrectly suggests that web components must be associated with web page elements, which Mr. Weadock refers to as "UI objects that can be displayed on a device."[113] As a preliminary matter, I disagree that the "UI objects" described in the specification are "web page elements." In the '755, '287, and '044 patents, the UI is for some unspecified mobile device and the UI objects are not (or at least need not) be parts of a web page. Indeed, the specification distinguishes "web pages" from "Application pages."[114]

104. Furthermore, while I certainly agree that web components *may* become associated with UI objects (indeed, the claims reflect the process by which this can occur, as illustrated in Figure 3F), this is not a definitional requirement of every web component. Otherwise, almost any UI object would be considered a "web component," since at some level, all UI objects serve the function of conveying information to the user. But treating "UI objects" and "web components" as virtual synonyms, as Mr. Weadock does, is inconsistent with how a POSA would understand the term and the claims of the '755, '287, and '044 patents, which refer to web components and UI objects as distinct claim elements and also describe the process by which web components in the web component registry *can* be associated with UI objects.[115]

[110] Exh. 1, Excerpt from 9,063,755 File History, U.S. Prov. Pat. Appl. No. 61/113,471, Nov. 11, 2008 at XMO_00002265-2290.

[111] *Id.* at XMO_00002266-2279.

[112] Weadock Decl. ¶43.

[113] *Id.*

[114] '755 patent at 12:4-10.

[115] *See, e.g.*, '755 patent at claim 1.

105. Mr. Weadock argues that Express Mobile's construction is supported by the '755 patent specification.[116] I disagree. None of these passages suggests that "web components" are functionality pre-associated with a web page element as Mr. Weadock appears to contend. Column 2, lines 33-34 of the '755 patent and related Figures 3E and 3F support Defendants' proposed construction, because they describe web components of web services that *can* be bound to UI objects, using the process described in the specification and reflected in the claims, as discussed above. The same is true of the other portions of the specification that Mr. Weadock references.[117]

106. In sum, a POSA would recognize that there is no requirement for web components to be pre-associated with particular UI objects as Express Mobile's construction suggests. Instead, the two requirements for a component to be considered a "web component" are that it be: 1) a software object that provides functionalities; and 2) that those functionalities are for a web service. A POSA would recognize that Defendants' proposed construction accurately encompasses both requirements, whereas Express Mobile's proposed construction does not.

107. This possibility bears no relationship to the discussion and disclosure of web components in the specification.[118] The addition of "associated with one or more web page elements to be displayed on a device" in Express Mobile's construction is also superfluous, as the claim language itself discloses how the web components are associated with UI objects (i.e., elements to be displayed on a device).[119]

[116] *See* Weadock Decl. ¶ 43.

[117] *See* Weadock Decl., ¶ 43 (citing '755 patent at 8:22-26, 22:15-17, 22:40–43, 25:6–15).

[118] *See, e.g.*, Exh. 1, Excerpt from 9,063,755 File History, U.S. Prov. Pat. Appl. No. 61/113,471, Nov. 11, 2008 at XMO_00002265-2290;
'755 patent at Fig. 3F and associated text; *id.* at 14:14–21, 6:31–47, 9:44–49, 10:4–11, 22:14–27, 22:40–43.

[119] *See, e.g.*, '755 patent at claim 1 ("associate the selected symbolic name with the defined UI object"); *id.* at claim 12 (similar); '287 patent at claims 1 and 15 (similar); '044 patent at claims 1 and 15 (similar).

F. Each symbolic name has an associated data format class type corresponding to a subclass of User Interface (UI) objects that support the data format of the symbolic name

Express Mobile's Proposed Construction	Defendants' Proposed Construction
No construction necessary	Indefinite[120]

108. Each of the asserted claims of the '287 patent and '044 patent require "symbolic names required for evoking one or more web components," where "each symbolic name has an associated data format class type corresponding to a subclass of User Interface (UI) objects that support the data format of the symbolic name."[121] It is my opinion that the scope of the "each symbolic name . . ." claim term would not be reasonably certain to a POSA based on the claims, the specification, and the file history because of the "data format class type" and "subclass of User Interface (UI) objects" elements.

109. First, the term "data format class type" does not appear anywhere in the specification and is not a term of art that a POSA would have been familiar with. The term appears to have been introduced to the claims, through a preliminary amendment filed on July 16, 2015 in the application that issued as the '287 patent.[122] In that amendment, the applicant provided limited remarks regarding the amendment and did not address the addition of the term "data format class type," except to state that "other amendments. . . are identical to the amendment to the claims made in the issued parent of this application, US Patent No. 9,063,755."[123] The applicant also alleged, without any specific citation, that "[s]upport for these amendments may be found in the file history of that patent."[124]

110. In forming my opinions expressed herein, I have reviewed the file history of the '755 patent and do not see any reference to "data format class type" in any claims of the '755 patent, nor does there appear to have been any discussion of this term during prosecution of that of the application that issued as the '755 patent. Indeed, based on my review of the '755 patent file history, the term "data

[120] I understand that the SAP Defendants do not join in this construction.

[121] '044 patent at 37:57–60; '287 patent at 37:51–61.

[122] Exh. 18, Excerpt from 6,546,397 File History, U.S. Pat. Appl. No. 09/454,061 at 3 (XMO_00000413).

[123] Exh. 18, Excerpt from 6,546,397 File History, U.S. Pat. Appl. No. 09/454,061 at 8 (XMO_00000418).

[124] *Id.*

format class type" is not used at all. Thus, it is not clear to me what the factual basis was for the applicant's representations made during prosecution of the '287 patent.

111. The specification of the '287 patent does refer in passing to "data types,"[125] but does not reference "data formats," "data classes," or "data format class types" at all.

112. In the file history of the '755 patent, which applicant generally pointed to as support for the claim amendment, there are several references to a "data format"[126], though the applicant did not discuss what was meant by this term in any detail. Moreover, just as there is no discussion of "data classes" or "data format class types" in the specification of the '287 patent, there is no discuss of what constitutes a "class" or "class type" for a given data format.

113. A POSA would recognize that "class" generally refers to a set of things that some property have in common. But it would not be clear what a "class type" of "data formats" refers to. The patent specification provides no guidance in this regard and a POSA would not consider the "data types" referenced in Table 1 to belong to different "classes" of "data formats" at all. To the extent any of these data types would be considered to have any features in common with one or more of the other data types, the specification of the '287 and '044 patents does not provide any guidance regarding what "classes" the data types should be divided into.

114. As a result, the scope of the claim requirement that "each symbolic name has an associated data format class type" would not be reasonably certain to a POSA based on the claims, the specification, and the file history. These materials to do not provide any guidance on how to determine the boundaries of a data format class type and thus, it is not possible to determine whether each symbolic name does, in fact, have an associated data format class type as the asserted claims require. For these reasons, it is my opinion that the term "data format class type" renders the asserted claims of the '287 patent and '044 patent indefinite.

[125] '287 patent at Table I, col. 15-16, 34:38–40, 34:55–58.

[126] Exh. 5, Excerpt from 9,063,755 File History, Reply "B" Under 37 C.F.R. §1.116(e), May 30, 2013 at 5-6 (XMO_00002852-2853); Exh. 6, Excerpt from 9,063,755 File History, Response to Office Action After Final Under 37 C.F.R §1.116 and Request for Continued Examination (RCE) Under 37 C.F.R. 1.114, Sept. 26, 2013 at 11 (XMO_00002877).

115. Second, the phrase "subclass of User Interface (UI) objects" separately renders the asserted claims indefinite because a POSA would not be able to determine the scope of the phrase with any reasonable certainty.

116. I note that the asserted claims do not refer to any specific class of UI objects and therefore, a POSA would not be able to determine whether a group of UI objects sharing some common attributes would be considered a class of UI objects, a subclass of some larger potential class, or simply a group of UI objects that share some attributes, but not enough to be considered a class or subclass.

117. As with the term "data format class type," the term "subclass" and the larger phrase "subclass of User Interface (UI) objects" are not used in the specification or file history of the patents. The term was introduced via the same July 16, 2015 preliminary amendment that added the term "data format class type," and similarly, no explanation or specific support for the term was provided.

118. The specification does discuss Java classes, from a Java class library[127], but a POSA would not understand these disclosures to be relevant to classes of UI objects or subclasses thereof. The specification of the patents does discuss various types of UI objects, e.g., slide show, video, image, etc. (*see, e.g.*, Table 1), but it does not set forth any classes of UI objects or criteria for a POSA to use to determine whether any of the described UI objects belong to the same class or subclass.

119. As a result, a POSA would not be able to understand the scope of the claims with reasonable certainty, because he or she would be unable to evaluate whether a group of UI objects belongs to a "class" and the required "subclass" thereof. Thus, the POSA would not be able to evaluate whether the "subclass" has an associated "data format class type" as the claim requires. Accordingly, it is my opinion that the "subclass of User Interface (UI) objects" renders the asserted claims of the '287 patent and '044 patent indefinite.

120. Mr. Weadock argues that the scope of the claim term is not indefinite because "data format class type" refers to "data type" and the "subclass of User Interface (UI) objects that support the data format" are the UI Objects listed, for example, in the "Input Candidates" and "Output Candidates"

[127] *See, e.g.*, '287 patent at 33:2–4.

columns of Table 1.[128] But A POSA would have no way of knowing if this actually was what the inventors meant, or whether the claims cover something else.

121. With respect to "data format class type," Mr. Weadock's opinion would require rewriting the claims to remove reference to "format" and "class" and instead interpret the term "data format class type" to mean "data type." While that is one potential way to resolve the ambiguity in the claim, it is certainly not the only one, and nothing in the specification would guide a POSA on how to specifically resolve the ambiguity in the claims.

122. In addition, while Mr. Weadock provides an example of a hypothetical "subclass of UI objects,"[129] he fails to identify what constitutes a "class of UI objects" from which any alleged "subclass" is selected. If all the claims required were data format class types corresponding to certain UI objects as Mr. Weadock contends, there would be no need to use the "subclass of" language at all and Mr. Weadock's opinion effectively reads it out of the claim. Again, while this may be one way to potentially resolve the ambiguity in the claims, nothing would suggest to a POSA that it is the only way. Taken at face value, the phrase "subclass of UI Objects" means something different than just "UI objects," but the specification provides no guidance on what that difference might be. For this additional reason, I disagree that the boundaries of the claims in which this phrase appears would be clear to a POSA, based on the lack of guidance to the POSA regarding the proper scope or meaning of the claim term "data format class type."

G. preferred UI object

Express Mobile's Proposed Construction	Defendants' Proposed Construction
a UI object associated with a data type that is favored	a UI object associated with a data type that is favored over the other UI object candidates for that data type

123. It is my opinion that a POSA would construe "preferred UI object" to mean a UI object associated with a data type that is favored over the other UI object candidates for that data type.

[128] Weadock Decl. ¶ 46.

[129] Weadock Decl. ¶ 47.

124. I understand that in prior cases, Express Mobile agreed to Defendants' proposed construction for this term.[130] This agreed-upon construction is consistent with a POSA's understanding of what it means for something to be preferred: it is favored over something else. A POSA would recognize that Express Mobile's proposed construction, which refers to a "favored" UI object without comparison to any other UI objects makes little sense and is contrary to the commonly understood usage of the term "preferred."

125. The only support Mr. Weadock provides for his opinion that a UI object can be considered a "preferred" UI object even though it is the *only* object available, is Table I of the '287 patent.[131]

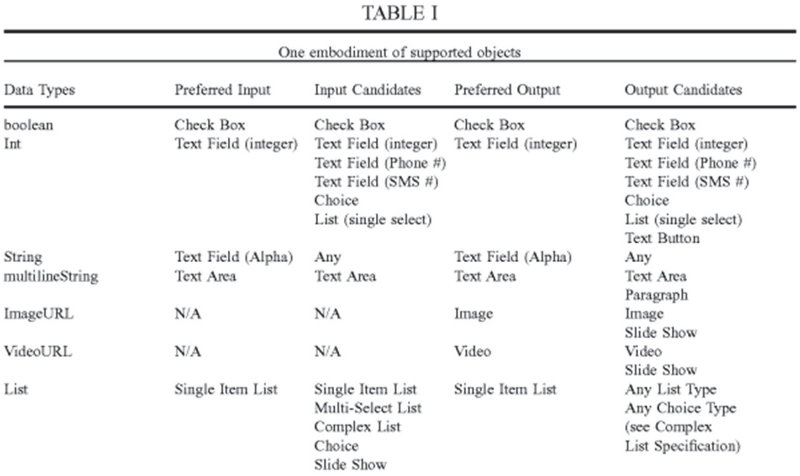


TABLE I

One embodiment of supported objects				
Data Types	Preferred Input	Input Candidates	Preferred Output	Output Candidates
boolean	Check Box	Check Box	Check Box	Check Box
Int	Text Field (integer)	Text Field (integer) Text Field (Phone #) Text Field (SMS #) Choice List (single select)	Text Field (integer)	Text Field (integer) Text Field (Phone #) Text Field (SMS #) Choice List (single select) Text Button
String	Text Field (Alpha)	Any	Text Field (Alpha)	Any
multilineString	Text Area	Text Area	Text Area	Text Area Paragraph
ImageURL	N/A	N/A	Image	Image Slide Show
VideoURL	N/A	N/A	Video	Video Slide Show
List	Single Item List	Single Item List Multi-Select List Complex List Choice Slide Show	Single Item List	Any List Type Any Choice Type (see Complex List Specification)

According to Mr. Weadock, Table I shows that a UI object may be considered a preferred UI object, even if it is the only object available for a given data type.[132]

126. But a POSA would recognize that Table I refers to "data types" and "Preferred Inputs" and "Preferred Outputs," whereas the claims of the '287 and '044 patent require "data format class types" with "preferred UI objects."[133] Thus, a POSA would not view Table I as a basis to broaden the scope of what it means for a UI object to be "preferred," particularly given additional claim language that requires the data format class type to correspond to a "subclass of UI objects," i.e., more than one UI object. A

[130] *See* Exh. 33, Joint Claim Construction Statement at 2, ECF No. 66, *Express Mobile, Inc. v. Expedia, Inc., et al*, Case No. 6:20-cv-801 (W.D. Tex. July 26, 2021) (agreeing that "preferred UI object" should be construed to mean "A UI object associated with a data type that is favored over the other UI object candidates for that data type.").

[131] Weadock Decl. ¶ 49.

[132] *Id.*; '287 Patent Table I.

[133] *Compare* '287 patent at Table I (cols. 15-16) *with* '287 patent at claim 1.

POSA would recognize that the claim requires the "preferred UI object" to be "selected," which also suggests there is a larger group of UI objects from which the preferred UI object is selected.[134]

I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct.

Executed on 12 Nov 2021, 2021 at Winchester, Massachusetts.

CHRISTOPHER SCHMANDT

[134] '287 patent at 37:57–61, 38:2–6, 39:31–35, 39:36–41; '044 patent at 37:57–61, 38:2–7, 39:25–30, 39:35–40.